This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 June 2002 (06.06.2002)

PCT

(10) International Publication Number
WO 02/45380 A2

(51) International Patent Classification[7]: H04L 29/06, G06F 1/00

(21) International Application Number: PCT/US01/45275

(22) International Filing Date:
30 November 2001 (30.11.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/250,261 30 November 2000 (30.11.2000) US
60/265,194 31 January 2001 (31.01.2001) US

(71) Applicant *(for all designated States except US)*: LANCOPE, INC. [US/US]; 1070 Greenway, Atlanta, GA 30305 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: COPELAND, John, A. III [US/US]; 1070 Greenway, Atlanta, GA 30305 (US).

(74) Agent: HARRIS, John, R.; Morris, Manning & Martin, LLP, 1600 Atlanta Financial Center, 3343 Peachtree Road, N.E., Atlanta, GA 3032601944 (US).

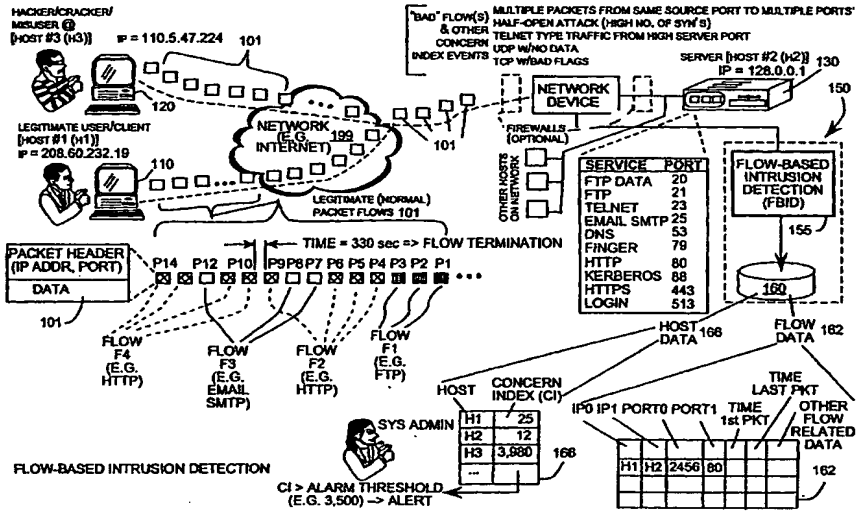
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv)) for US only*

[Continued on next page]

(54) Title: FLOW-BASED DETECTION OF NETWORK INTRUSIONS



(57) Abstract: A flow-based intrusion detection system for detecting intrusions in computer communication networks. Data packets representing communications between hosts in a computer-to-computer communication network are processed and assigned to various client/server flows. Statistics are collected for each flow. Then, the flow statistics are analyzed to determine if the flow appears to be legitimate traffic or possible suspicious activity. A concern index value is assigned to each flow that appears suspicious. By assigning a value to each flow that appears suspicious and adding that value to the total concern index of the responsible host, it is possible to identify hosts that are engaged in intrusion activity. When the concern index value of a host exceeds a preset alarm value, an alert is issued and appropriate action can be taken.

WO 02/45380 A2

FLOW-BASED DETECTION OF NETWORK INTRUSIONS

CROSS REFERENCE TO RELATED APPLICATIONS

This Patent Application claims priority to the United States provisional patent application serial no. 60/250,261 entitled "System and Method for Monitoring Network Traffic" filed November 30, 2000 and United States provisional patent application serial no. 60/265,194 entitled "The Use of Flows to Analyze Network Traffic" filed on January 3, 2001, both of which are incorporated in their entirety by reference and made a part hereof.

REFERENCE TO COMPUTER PROGRAM LISTING SUBMITTED ON CD

This application incorporates by reference the computer program listing appendix submitted on (1) CD-ROM entitled "Flow-Based Engine Computer Program Listing" in accordance with 37 C.F.R. § 1.52(e). Pursuant to 37 C.F.R. § 1.77(b)(4), the material on said CD-ROM is incorporated by reference herein, said material being identified as follows:

Sizein Bytes	Date of Creation	File Name
154,450	November 30, 2001	LANcope Code.txt

traffic designed bypass the blocking rules established. Additionally, almost all commercially available IDS are signature based detection systems or anomaly based systems.

Signature based detection systems piece together the packets in a connection to collect a stream of bytes being transmitted. The stream is then analyzed for certain strings of characters in the data commonly referred to as "signatures." These signatures are particular strings that have been discovered in known exploits. The more signatures that are stored in a database, the longer it takes to do on exhaustive search on each data stream. For larger networks with massive amounts of data transferred, a string comparison approach is unfeasible. Substantial computing resources are needed to analyze all of the communication traffic.

Besides, even if a known exploit signature has been discovered, the signature is not useful until it is has been installed and is available to the network. In addition, signature analysis only protects a system from known attacks. Yet, new attacks are being implemented all the time. Unfortunately, a signature based detection system would not detect these new attacks and leave the network vulnerable.

Another approach to intrusion detection includes detection of unusual deviation from normal data traffic commonly referred to as "anomalies." Like signature-based detection systems, many current anomaly based intrusion detection systems only detect known methods of attacks. Some of these known anomaly based attacks include TCP/IP stack fingerprinting, half-open attacks, and port scanning. However, systems relying on known attacks are easy to circumnavigate and leave the system vulnerable. In addition, some abnormal network traffic happens routinely, often non-maliciously, in normal network traffic. For example, an incorrectly entered address could be sent to an unauthorized port and be interpreted as an abnormality. Consequently, known anomaly based systems tend to generate an undesirable number of false alarms which creates a tendency to have all alarms generated to become ignored.

Some known intrusion detection systems have tried to detect statistical anomalies. The approach is to measure a baseline and then trigger an alarm when deviation is detected. For example, if a system typically has no traffic from individual workstations at 2 am, activity during this time frame would be considered suspicious. However, baseline systems have typically been ineffective because the small amount of malicious activity is

be legitimate traffic or possible suspicious activity. A value, referred to as a "concern index," is assigned to each flow that appears suspicious. By assigning a value to each flow that appears suspicious and adding that value to an accumulated concern index associated with the responsible host, it is possible to identify hosts that are engaged in intruder activity without generation of significant unwarranted false alarms. When the concern index value of a host exceeds a preset alarm value, an alert is issued and appropriate action can be taken.

Generally speaking, the intrusion detection system analyzes network communication traffic for potential detrimental activity. The system collects flow data from packet headers between two hosts or Internet Protocol (IP) addresses. Collecting flow data from packet headers associated with a single service where at least one port remains constant allows for more efficient analysis of the flow data. The collected flow data is analyzed to assign a concern index value to the flow based upon a probability that the flow was not normal for data communications. A host list is maintained containing an accumulated concern index derived from the flows associated with the host. Once the accumulated concern index has exceeded an alarm threshold value, an alarm signal is generated.

BRIEF DESCRIPTION OF THE DRAWINGS

Benefits and further features of the present invention will be apparent from a detailed description of preferred embodiment thereof taken in conjunction with the following drawings, wherein like elements are referred to with like reference numbers, and wherein:

FIG. 1 is a functional block diagram illustrating a flow-based intrusion detection system constructed in accordance with a preferred embodiment of the present invention.

FIG. 2 is a diagram illustrating headers of datagrams.

Overview

Turning to the figures, in which like numerals indicate like elements throughout the several figures, FIG. 1 provides an overview of a flow-based intrusion detection system or engine 155 in accordance with an exemplary embodiment of the present invention. The flow-based intrusion detection system 155 monitors network computer communications. The network computer communications are routed via a known global computer network commonly known as the Internet 199. In accordance with an aspect of the invention, the intrusion detection engine 155 is incorporated into a monitoring appliance 150, together with a database 160 that stores information utilized in the intrusion detection methodology.

The operating environment of the intrusion detection system 155 is contemplated to have numerous hosts connected by the Internet 199, e.g. Host #1, Host #2, Host #3 (also referred to as H1-H3 respectively). Hosts are any computers that have full two-way access to other computers on the Internet 199 and have their own unique IP address. For example Host #1 has an exemplary IP address of 208.60.239.19. The Internet 199 connects clients 110 with a host server 130 in known client/server relationship.

In a typical configuration, some computers are referred to as "servers', while others are referred to as "clients." A server computer such as Host #2 130 typically provides responses to requests from client computers and provides services, data, resources, and the like. While a client computer such as Host #1 110 typically requests and utilizes the services, data, resources, and the like provided by the server.

It is known in the art to send communications between hosts via the Internet 199. The Internet Protocol (IP) is the method by which data is sent from one host computer to another on the Internet 199. Each host on the Internet 199 has an IP address that uniquely identifies it from all other computers. When data is transmitted, the message gets divided into packets 101. Packets 101 are discussed in more detail in reference to FIG. 2.

Each IP packet 101 includes a header that contains both the sender's Internet address and receiver's Internet address. The packets 101 are forwarded to the computer whose address is specified. Illustrated is a legitimate user/client 110, host #1 (H1), with an IP address of 208.60.239.19 and a server, host #2 (H2), with an IP address of 128.0.0.1.

As shown, a client 110 communications with a server 130 by sending packets 101 of data. A packet 101 is a unit of data that is routed between an origin and destination. As

host server port 80. File Transfer Protocol (FTP) control communications are sent to the server port 21, while FTP data transfer originates from port 20. The FINGER service utilizes service contact port 79, the domain name service (DNS) utilizes service contact port 53, and Telnet communications utilize service contact port 23. As illustrated, common services are typically associated with specific predetermined service contact ports.

Also illustrated in FIG. 1 are four flows, F1 through F4, between by client host #1 110 and service host #2 130. Flow F1 is a file transfer utilizing the File Transfer Protocol (FTP). As shown, the file transfer (flow F1) is delivered by a stream of packets 101 (P1—P3) that will be reassembled by the receiving host 110.

After the file transfer is completed, the client 110 initiates an HTTP Web session (flow F2) with server 120. Those skilled in the art understand that a Web session typically occurs when an Internet browser computer program such as MICROSOFT INTERNET EXPLORER or NETSCAPE NAVIGATOR requests a web page from a World Wide Web (WWW) service on port 80. Packets P4, P5, P6, and P9 are associated with the Web traffic of flow F2. These packets may contain data such as a JPG format picture to be displayed, text, a JAVA program, or other informational materials to be displayed or handled by the client's Internet browser program.

Continuing the example of FIG. 1, while the web session of flow F2 is still open, the client 110 sent an email illustrated by flow F3. As shown, the email packets of flow F3 may be interleaved with the previously opened Web session of flow F2. As illustrated, packets P7, P8, and P12 contain the e-mail message.

Finally, the client 110 requests another web page from the server 120, initiating yet another HTTP flow F4. Packets P9, P10, P11, P12, and P14 represent the new Web traffic.

In accordance with an aspect of the invention, a flow is considered terminated after a predetermined period of time has elapsed on a particular connection or port. For example, if HTTP Web traffic on port 80 ceases for a predetermined period of time, but other traffic begins to occur on port 80 after the expiration of that predetermined time period, it is considered that a new flow has begun, and the system responds accordingly to assign a new flow number and track the statistics and characteristics thereof. In the

Consequently, abnormal flows and/or events identified by the intrusion detection engine 155 will raise the concern index (CI) for the associated host. The intrusion detection engine 155 analyzes the data flow between IP devices. However, different types of services have different flow characteristics associated with that service. Therefore, a C/S flow can be determined by the packets exchanged between the two hosts dealing with the same service.

In accordance with an aspect of the invention, the intrusion detection engine 155 works by assigning data packets 101 to various flows. The engine 155 collects information about and statistics associated with each flow and stores this information and statistics in a database 160. The flow database 160 comprises a flow data structure 162 and a host data structure 166. The flow data structure 162 stores collected flow information such as the IP addresses. The engine determines which host has a lower IP address and assigns that host IP0. The other host is assigned IP1. Port0 is associated with IP0 and port1 is the service connection port for host1. The flow data structure 162 also stores time and other related packet information derived from the packet header. In the disclosed embodiment, this time information (e.g. time of the first packet, time of the last packet) is utilized to measure the elapse of time for purposes of flow delimiting, as described above.

The intrusion detection engine 155 analyzes the flow data 160 to determine if the flow appears to be legitimate traffic or possible suspicious activity. Flows with suspicious activity are assigned a predetermined concern index (CI) value based upon a heuristically predetermined assessment of the significance of the threat of the particular traffic or flow or suspicious activity. The flow concern index values have been derived heuristically from extensive network traffic analysis. Concern index values are associated with particular hosts and stored in the host data structure 166 (FIG. 1). Exemplary concern index values for various exemplary flow-based events and other types of events are illustrated in connection with FIGS. 6 and 7.

By assigning a value to each flow that appears suspicious and adding that value to a total CI of the host responsible for the flow, it is possible to identify hosts that are engaged in intruder activities. When the CI of a host exceeds a preset alarm threshold, a alarm signal may be generated. In the example of FIG. 1, host H3 has an accumulated CI of 3,980. This exceeds the preset threshold of 3,500 for that network and a system

reliance on earlier exchanges between the source and destination computer. Packets 101 have a header and a data segment as illustrated by FIG. 2. The term "packet" in present-day parlance has generally replaced the term "datagram".

Restated, a packet 101 is the unit of data that is routed between an origin and destination on a packet-switched network such as the Internet 199. A packet-switching scheme is an efficient method of handling transmissions on a connectionless network. However, connection-oriented protocols can be utilized to create a session. A session is a series of interactions between two communication end points that occur during the span of a single connection. A detailed discussion of a TCP/IP session is described in reference to FIG. 3. However, a host can send a message without establishing a connection with the recipient. That is, the host simply sends a packet 101 onto the network 199 with the destination address and hopes that it arrives.

FIG. 2 illustrates an exemplary TCP/IP packet or datagram 210 and an exemplary UDP datagram 240. In a typical TCP/IP packet like 210, each packet typically includes a header portion comprising an IP header 220 and a TCP header 230, followed by a data portion that contains the information to be communicated in the packet. The information in the IP header 220 contained in a TCP/IP packet 210, or any other IP packet, contains the IP addresses and assures that the packet is delivered to the right host. The transport layer protocol (TCP) header follows the Internet protocol header and specifies the port numbers for the associated service.

The header portion in the typical TCP/IP datagram 210 is 40 bytes including 20 bytes of IP header 220 information and 20 bytes of TCP header 230 information. The data portion or segment associated with the packet 210 follows the header information.

In regards to a typical IP packet 210, the first 4 bits of the IP header 220 identify the Internet protocol (IP) version. The following 4 bits identify the IP header length in 32 bit words. The next 8 bits differentiate the type of service by describing how the packet should be handled in transit. The following 16 bits convey the total packet length.

Large packets tend to be fragmented by networks that cannot handle a large packet size. A 16-bit packet identification is used to reassemble fragmented packets. Three one-bit set of fragmentation flags control whether a packet is or may be fragmented. The 13-bit fragment offset is a sequence number for the 4-byte words in the packet when reassembled. In a series of fragments, the first offset will be zero.

underlying protocol. The UDP protocol is transaction oriented and delivery protection is not guaranteed. Applications requiring reliable delivery of data typically use the previously described Transmission Control Protocol (TCP).

The 16-bit UDP source port is a field to which port a reply, when meaningful, should be addressed. The 16-bit UDP destination port specifies the server program on the receiving host to execute the packet. Next, the 16-bit UDP message length field is the length in bytes of the user datagram including header and any data. Following the length field is the 16-bit checksum of the UDP header, the UDP pseudo header information 250 from an IP header 220, and the data.

As will be understood by those skilled in the art, the fundamental Internet service consists of a packet delivery system. Internet service is typically considered "connectionless" because each packet is treated independently of all others. Some transport protocols such as UDP provide unreliable service because the delivery of the packet is not guaranteed. Other transport protocols such as TCP provide a mechanism to ensure delivery of a packet and therefore can be used to establish computer-to-computer "sessions" in the conventional sense of the term. FIG. 3 illustrates a typical TCP/IP session and the guaranteed packet delivery mechanism.

As previously stated, the flow-based engine 155 does not analyze the data segments of packets for signature identification. Instead, the engine 155 associates all packets with a flow. It analyzes certain statistical data and assigns a concern index value to abnormal activity. The engine 155 builds a concern index for suspicious hosts by detecting suspicious activities on the network. An alarm is generated when those hosts build enough concern (in the form of a cumulated CI value) to cross the network administrator's predetermined threshold.

Session

Turning next to FIG. 3, a TCP session 300 is a full duplex connection that allows concurrent transfer of data in both directions. Before the transfer can start, both the sending and receiving application programs interact with their respective operating systems, informing them of the impending stream transfer. Protocol software communicates by sending messages across, verifying that the transfer is authorized, and indicating that both sides are ready to receive data.

underlying protocol. The UDP protocol is transaction oriented and delivery protection is not guaranteed. Applications requiring reliable delivery of data typically use the previously described Transmission Control Protocol (TCP).

The 16-bit UDP source port is a field to which port a reply, when meaningful, should be addressed. The 16-bit UDP destination port specifies the server program on the receiving host to execute the packet. Next, the 16-bit UDP message length field is the length in bytes of the user datagram including header and any data. Following the length field is the 16-bit checksum of the UDP header, the UDP pseudo header information 250 from an IP header 220, and the data.

As will be understood by those skilled in the art, the fundamental Internet service consists of a packet delivery system. Internet service is typically considered "connectionless" because each packet is treated independently of all others. Some transport protocols such as UDP provide unreliable service because the delivery of the packet is not guaranteed. Other transport protocols such as TCP provide a mechanism to ensure delivery of a packet and therefore can be used to establish computer-to-computer "sessions" in the conventional sense of the term. FIG. 3 illustrates a typical TCP/IP session and the guaranteed packet delivery mechanism.

As previously stated, the flow-based engine 155 does not analyze the data segments of packets for signature identification. Instead, the engine 155 associates all packets with a flow. It analyzes certain statistical data and assigns a concern index value to abnormal activity. The engine 155 builds a concern index for suspicious hosts by detecting suspicious activities on the network. An alarm is generated when those hosts build enough concern (in the form of a cumulated CI value) to cross the network administrator's predetermined threshold.

Session

Turning next to FIG. 3, a TCP session 300 is a full duplex connection that allows concurrent transfer of data in both directions. Before the transfer can start, both the sending and receiving application programs interact with their respective operating systems, informing them of the impending stream transfer. Protocol software communicates by sending messages across, verifying that the transfer is authorized, and indicating that both sides are ready to receive data.

Alternatively, a host may desire to keep a session active even after if has finished sending its current data. If more data is to be sent in the near future, it is more efficient to keep a session open than it is to open multiple sessions. A session wherein the connection is kept open in case future data is to be communicated is typically referred to as a "persistent" session. In this scenario, a session is closed by sending a packet with the reset flag (R) set (also called a "reset packet") after no data is delivered after a period of time. Many browser applications provide a 300-second window of inactivity before closing a session with an R packet (reset).

The described TCP session 300 of FIG. 3 is a generic TCP session in which a network might engage. In accordance with the invention, flow data is collected about the session to help determine if the communication is abnormal. In the preferred embodiment, information such as the total number of packets sent, the total amount of data sent, the session start time and duration, and the TCP flags set in all of the packets, are collected, stored in the database 160, and analyzed to determine if the communication was suspicious. If a communication is deemed suspicious, i.e. it meets predetermined criteria, a predetermined concern index value associated with a determined category of suspicious activity is added to the cumulated CI value associated with the host that made the communication.

For example, a TCP/IP packet with both the SYN flag and the FIN flag set would not exist in a normal communication. Because a packet with both the SYN and FIN flags set is undefined, each operating system handles this packet in different methods. An operating system may send an ICMP message, a reset, or possibly just ignore it and send nothing. Consequently, an intruder may send a SYN-FIN packet specifically to help identify the operating system of the targeted host.

As another example, if a particular host sends a large number of SYN packets to a target host and in response receives numerous R packets from the targeted host, a potential TCP probe is indicated. Likewise, numerous UDP packets sent from one host to a targeted host and numerous ICMP "port unavailable" packets received from the targeted host indicates a potential UDP probe. A stealth probe is indicated by multiple packets from the same source port number sent to different port numbers on a targeted host.

As has been described elsewhere, UDP packets are often used in connection with streaming media and other applications that provide data to many hosts. A UDP packet

services is analyzed separately. Therefore, as defined, the illustrated communications represent three distinct flows.

The first flow illustrated would be Web traffic (HTTP protocol) between the client at IP ADDRESS1 and the server at IP ADDRESS0. The client Web browser opens a random ephemeral high port (51,132) as illustrated in the example. A high port is utilized because the low port numbers less than 1024 are preassigned for designated services. One these designated services is port 80 for HTTP, which transfers displayable Web pages and related files in the known manner. The Web browser sends the request to the server's port 80. The server port responds by sending the requested Web page data in packets wherein the port number in the packets transmitted to the client sets the destination port to 51,132 of the client. All communications by clients utilizing HTTP is sent to port 80 of the server. One C/S flow would be the HTTP communications between port 51,132 of ADDRESS1 and port 80 of ADDRESS0.

A flow is terminated if no communications occur between the two IP addresses and the one low port (e.g. port 80) for 330 seconds. Most Web browsers or a TCP connection send a reset packet (i.e. a packet with the R flag set) if no communications are sent or received for 5 minutes. An analysis can determine if the flow is abnormal or not for HTTP communications.

The next flow illustrated is email traffic between the client and server utilizing port 25. The client email application opens a random high ephemeral port, e.g. port 49,948 as illustrated in FIG. 4. The client's email application sends the email utilizing the Simple Mail Transfer Protocol (SMTP) to the server's port 25. Port 25 is conventionally designated for SMTP communications. A flow is terminated if no communications are delivered between the two IP addresses and the low port for 330 seconds. If the client sends another SMTP email packet or packets within 330 seconds of the end of the first email to the server, only one flow would exist.

For example, as shown in FIG. 4, if a second email packet originating from the ephemeral port 35,620 is sent within 330 seconds, only one flow would exist. If the second email packet was later than 330 seconds from the first sent email, it would be classified as another flow for analysis purposes. An analysis can determine if the flow is abnormal or not for SMTP communications.

understood that the terms "computer," "operating system," and "application program" include all types of computers and the program modules designed to be implemented by the computers.

The discussion of methods that follow, especially in the software architecture, is represented largely in terms of processes and symbolic representations of operations by conventional computer components, including a central processing unit (CPU), memory storage devices for the CPU, network communication interfaces, connected display devices, and input devices. Furthermore, these processes and operations may utilize conventional computer components in a heterogeneous distributed computing environment, including remote file servers, remote computer servers, and remote memory storage devices. Each of these conventional distributed computing components is accessible by the CPU via a communication network.

The processes and operations performed by the computer include the manipulation of signals by a CPU, or remote server such as an Internet Web site, and the maintenance of these signals within data structures reside in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical, optical, or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to effectively convey teachings and discoveries to others skilled in the art. For the purposes of this discussion, a process is understood to include a sequence of computer-executed steps leading to a concrete, useful, and tangible result, namely, the detection of intruders based upon C/S flows and other activity deemed heuristically to be a threat substantial enough to warrant assignment of a concern index value.

These steps generally require manipulations of quantities such as IP addresses, packet length, header length, start times, end times, port numbers, and other packet related information. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is conventional for those skilled in the art to refer to these signals as bits, bytes, words, values, elements, symbols, characters, terms, numbers, points, records, objects, images, files or the like. It should be kept in mind, however, that these and similar terms should be associated with appropriate quantities for computer

client and server network applications that are being operating by the hosts that are observed participating in the flows observed (port profiling).

Packet Classifier

The header data is read by the packet classifier thread 510. The packet classifier thread 510 runs whenever new packet information is available. Based on the source and destination IP addresses, the thread 510 searches for an existing flow in the flow data structure 162. To facilitate searching and record insertion, a symmetric hash of the two IP addresses is generated and used as the index of an array that points to the beginning of a two-way linked list of all flows with that hash value. As known to those skilled in the art, a symmetric hash is a mathematical process that creates a probabilistically unique number that facilitates rapid indexing and sorting within a data structure such as flow data structure 162.

Flow processing is done for TCP and UDP packets, and the port numbers in the transport layer header are used to identify the flow record to be updated. For ICMP packets that constitute rejections of a packet, the copy of the rejected packet in the ICMP data field is used to identify the IP addresses and port numbers of the corresponding flow.

For purposes of the description which follows, the IP address with the lower value, when considered as a 32-bit unsigned integer, is designated ip[0] and the corresponding port number is designated pt[0]. The higher IP address is designated ip[1] and the corresponding TCP or UDP port number is designated pt[1]. At some point, either pt[0] or pt[1] may be designated the "server" port by setting an appropriate bit in a bit map that is part of the flow record (record "state", bit 1 or 2 is set).

If a particular packet 101 being processed by the packet classifier 510 matches a particular entry or record in the flow data structure 162, data from that particular packet 101 is used to update the statistics in the corresponding flow data structure record. A packet 101 is considered to match to a flow data structure record if both IP numbers match and:

a) both port numbers match and no port is marked as the "server" port, or

b) the port number previously marked as the "server" port matches, or

```
unsigned char flag[2][7];//0 bad, 1 reset, 2 urgent, 3 syn, 4 syn-ack, 5 fin, 6
        fragments, // (counts of packets seen with various TCP flag combinations)
-7      UDP  rejects
unsigned short scans ; // max number ports seen for ip pair, detects "Port Scans"
        } flow[SLOTS] ;
```

Notice that many of the fields are counters for each host, e.g., the number of packets and bytes sent, the number of packets with various TCP flag-bit combinations sent for TCP flows, the number of ICMP "port-unavailables" for a UDP flow. Also bitmaps can be filled in, such as the bitmap of all TCP flags seen which has been bitwise OR'ed with the TCP flag field of each TCP packet. Data is filled in for the source (originating) host.

The packet classifier thread 510 also adds some data directly to the host data structure 166. Most of this data could be added later by the flow collector thread 520 (such as bytes sent by each host), but adding it on a packet by packet basis allows collection of real time rate information (such as bytes sent in each time interval). These records are indicated in the host data structure 166 below.

Host Data Structure

The host data structure 166 accumulates data on all hosts that are observed participating in a flow. A description of this data structure in C language format follows:

```
#define HOST_SLOTS  65537  // number Host slots
struct host_db {
  // data added by the Packet Classifier Thread
        unsigned long ip ;              //ip address
        unsigned long down ; // linked list index
        unsigned long up;      // linked list index
        unsigned long start ;  // time host record started
        unsigned long last ;  // time of last packet from this host
        unsigned long udp_bytes ;  // UDP bytes sent and received
        unsigned long bytes_in ;     // bytes received
        unsigned long bytes_in_pp ;  // Bytes over last 5 min interval
```

and a logic-tree analysis is done to classify them as either a normal flow, or a potential probe or other suspicious activity warranting assignment of a concern index value.

Normal flows are those for which the corresponding statistics indicate a normal exchange of information between two hosts. The host that initiated the flow is considered the client (i.e. the computer that sent TCP SYN packets or sent an initial UDP packet). The other host is considered the server (i.e. the computer that sent TCP SYN-ACK packets or responded to a UDP packet). Some data is exchanged during a normal flow.

A potential probe is a flow that appears to have one host (a possible intruder) sending packets to gain information about another host (an intended victim). An example of a potential probe is a flow that has TCP packets of any sort sent by one host (the intruder) and approximately the same number of TCP reset packets sent by the other. Another example is a flow which has UDP packets answered by ICMP "port unavailable" packets. A flow with ICMP "destination unreachable" packets sent by one host would be considered a potential probe being done by the other host.

In accordance with the invention, some potential probes are much more likely to indicate probing than others are. To handle this, a value called the "concern index" is calculated or otherwise determined for each flow, and this value is added to the concern index value being accumulated in the host data structure 166. Table I of FIG. 6 shows one scheme for assigning concern index values due to the flow analysis. After the flow is analyzed, the flow record is written to the flow log file and then cleared from the flow data structure.

Other Concern Index Increments

Concern index (CI) values calculated from packet anomalies also add to a host's accumulated concern index value. Table II of FIG. 7 shows one scheme for assigning concern index values due to other events revealed by the flow analysis. For example, there are many combinations of TCP flag bits that are rarely or never seen in valid TCP connections. When one of these combinations is recognized by the packet classifier thread 510, it directly adds a predetermined value to the sending host's accumulated concern index value. When the packet classifier thread 510 searches along the flow linked-list (i.e. flow data 162) for a match to the current packet 101, it keeps count of the number of flows active with matching IP addresses but no matching port number. If this number exceeds a

The alert manager 530 also looks for hosts whose CI or traffic (byte rate) exceeds preset alarm thresholds and which have not been handled on previous runs. The new alarm conditions can cause immediate operator notification by an operator notification process 542. These conditions can be highlighted on the user interface, and cause SNMP trap messages to be sent to a network monitor such as HP Openview, and/or email messages to the network administrator which in turn may cause messages to be sent to beepers or cell phones. Messages can also be sent to cause automated devices such as a firewall manager 544 to drop packets going to or from an offending host. It will thus be appreciated that the present invention advantageously operates in conjunction with firewalls and other network security devices and processes to provide additional protection for an entity's computer network and computer resources.

Hardware

A preferred hardware configuration 800 of an embodiment that executes the functions of the above described flow-based engine is described in reference to FIG. 8. FIG. 8 illustrates a typically hardware configuration 800 for a network intrusion detection system. A monitoring appliance 150 serves as a pass-by filter of network traffic. A network device 135, such as a router, switch, hub, tap, or the like, provides the location for connecting the monitoring appliance 150 to the network 899 for monitoring the network traffic.

As illustrated, the monitoring appliance 150 is preferably configured with two network interface cards (NIC) 830 such as 3COM brand model 932 10/100 MHz Ethernet adapters or other adapters to match the network. However, it should be apparent to one skilled in the art that one or more cards can be utilized to accomplish the functions of the presently described dual card system. The monitor NIC 834 is typically set to a promiscuous mode or a similar function. The promiscuous mode is a mode of operation in which every data packet passing through the network device 135 will be received and read. An admin NIC 838 allows network interfacing and handles commands sent from the monitoring appliance 135. A NIC driver 820 enables the network traffic data to be exchanged with the processor 850. Other drivers 825 are utilized to interface or communicate with other devices including peripherals. These peripherals include

As discussed previously, the header data if each packet processed is read by the packet classifier thread 510. Based on the source and destination IP addresses, the thread 510 searches for an existing flow in the flow data structure 162, which is embodied as a data array in memory. A symmetric hash of the two IP addresses is used as the index into the array that points to the beginning of a two-way linked list of all flows with that hash value.

Flow processing is done for TCP and UDP packets, and the port numbers in the transport layer header are used to identify the flow record to be updated. For ICMP packets that constitute rejections of a packet, the copy of the rejected packet in the ICMP data field is used to identify the IP addresses and port numbers of the corresponding flow.

A packet 101 is considered to match to a flow data structure record if both IP numbers match and:

a) both port numbers match and no port is marked as the "server" port, or

b) the port number previously marked as the "server" port matches, or

c) one of the port numbers matches, but the other does not, and the neither port number has been marked as the server port (in this case the matching port number is marked as the "server" port).

If a new flow is determined, the yes branch of step 914 is followed by step 916. In step 916, a new flow record is created. If no flow exists that matches the current packet, a new flow record is started using the IP addresses and port numbers from the current packet, and is linked to the end of the appropriate linked list of flow records.

The IP address with the lower value, when considered as a 32-bit unsigned integer, is designated ip[0] and the corresponding port number is designated pt[0]. The higher IP address is designated ip[1] and the corresponding TCP or UDP port number is designated pt[1]. At some point, either pt [0] or pt[1] may be designated the "server" port by setting a the appropriate bit in a bit map that is part of the flow record (record "state", bits 1 or 2 set).

Step 916 is followed by step 918, in which the flow records in the flow data structure 162 are updated. The time that the flow started, the packet capture time, is written into the record "start." The flow data structures updated by the packet classifier thread is discussed in detail in reference to FIG. 5. Step 918 is returned to step 912, in which the thread 510 determines if a new packet is available.

Step 945 is followed by step 946. In step 946, the flow record is written to the flow log file. Step 946 is followed by step 947. In step 947, the flow record is cleared from the flow data structure. After step 947, the thread is returned to step 942, in which the thread awaits for the requisite time.

Referring next to FIG. 9C, the alarm manager thread 530 begins with step 972. In step 972, the thread 530 determines if a periodic time has elapsed. If the requisite time period has not elapsed, the no branch of step 972 is followed to step 972, in which the thread 530 awaits the time to elapse.

If the time has elapsed, the yes branch of step 972 is followed to step 973, in which the thread 530 performs concern index search. The alert manager thread 530 runs periodically (e.g., following the flow manager thread 520) and does a linear search through the host data structure 166.

Step 973 is followed by step 974. In step 974, it compiles a number of lists that are written to various output files for use by the user interface programs. For example, it collects an alert list of hosts with CI above a certain threshold. This threshold may be adjusted so that the list is about 100 host records long. A user interface program will preferably sort this list and display, in order of descending CI value, the top 60 hosts with high CI values. A similar list based on average byte rate over the last time interval (e.g., 5 minutes) is also generated. If a range, or set of ranges, of IP addresses have been defined by the network administrator as "inside addresses," separate lists can be generated for "inside" and "outside" hosts. Numerous other queries and reports 548 can be generated for review and analysis by the network administrator. The alert manager thread 530 writes the updated data to various output files for use by the user interface, or for later off-line analysis.

Step 974 is followed by step 975, in which the thread 530 determines if an alarm threshold has been exceeded. If the alarm threshold has not been exceeded, the no branch of step 975 is returned to perform step 972. In step 972, the thread 530 determines if a requisite time period has elapsed.

If an alarm threshold has been exceeded, the yes branch of step 975 is followed to step 976. In step 976, the alert manager thread generates certain predetermined signals designed to drawn the attention of a system administrator or other interested person. The alert manager 530 looks for hosts whose CI or traffic (byte rate) exceeds preset alarm

CLAIMS

What is claimed is:

1. A method of analyzing network communication traffic for potential intrusion activity, comprising the steps of:

assigning packets to a flow;

collecting flow data from packet headers;

analyzing collected flow data to assign a concern index value to the flow based upon a probability that the flow was not normal for data communications;

maintaining an accumulated concern index from flows associated with a host; and

issuing an alarm signal once the accumulated concern index has exceeded an alarm threshold value.

2. The method of claim 1, wherein the flow consists of the packets exchanged between two hosts that are associated with a single service.

3. The method of claim 1, wherein the alarm signal updates a firewall for filtering packets transmitted by a host.

4. The method of claim 1, wherein the alarm signal generates a notification to the network administrator.

5. The method of claim 1, wherein each concern index value associated with a respective potential intrusion activity is a predetermined fixed value.

10. A system for analyzing network communication traffic, comprising:

a computer system operable to classify packets into flows, collect flow data from packet header information, analyze collected flow data to assign a concern index value wherein each concern index value associated with a respective potential intrusion activity is a predetermined fixed value, and generate an alarm signal; and

a communication system coupled to the computer system operable to send packets from one host to another host.

11. A system for analyzing network communication traffic, comprising:

a processor operable to classify packets into flows, collect flow data from packet header information, analyze collected flow data to assign a concern index value wherein each concern index value associated with a respective potential intrusion activity is a predetermined fixed value, and generate an alarm signal;

memory coupled to the processor operable to store the flow data;

a database coupled to processor operable to store log files; and

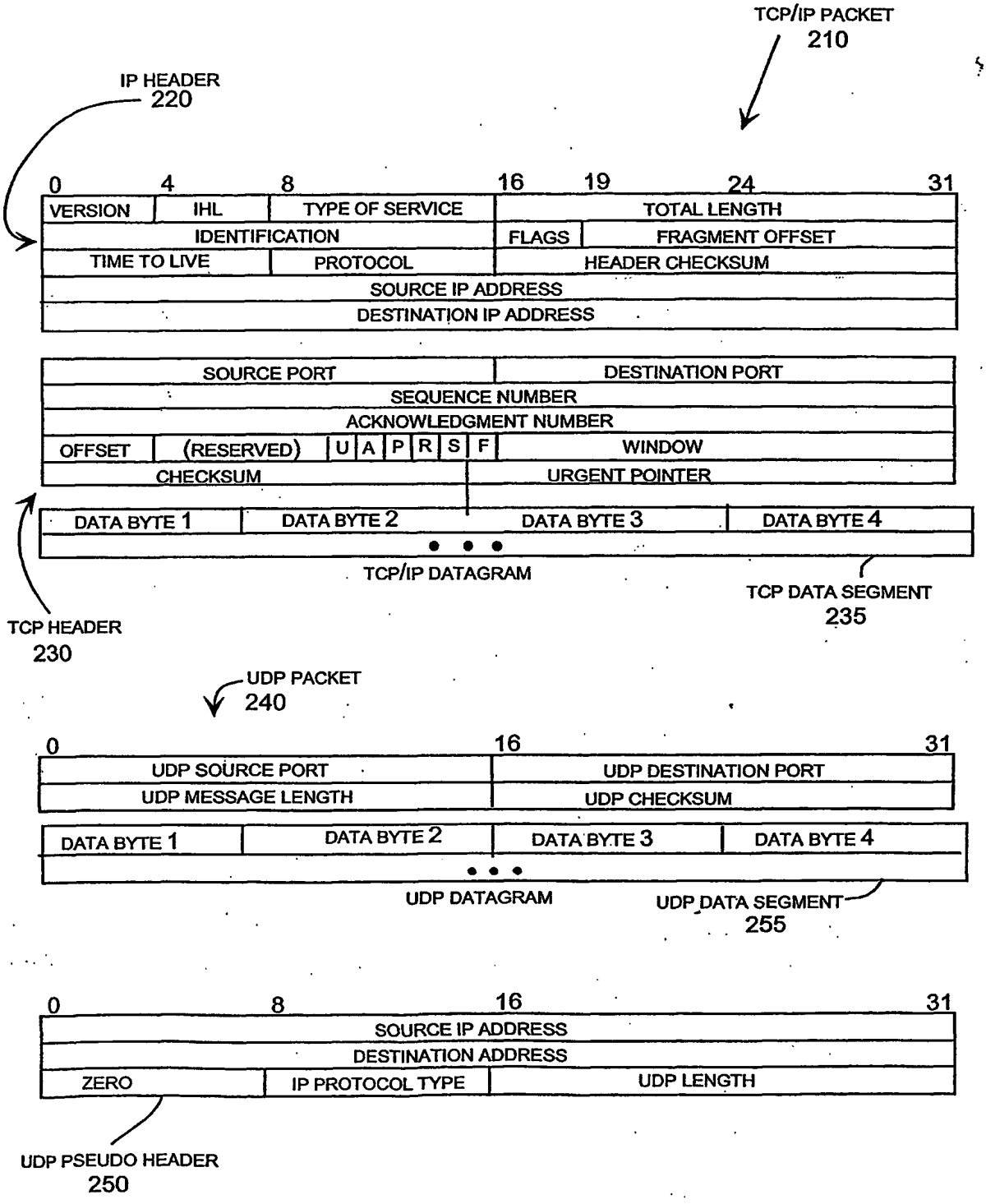
and a network interface coupled to the processor operable to monitor network traffic.

12. A method of analyzing network communication traffic for potential intrusion activity, comprising the steps of:

analyzing packet header information;

determining a transport level protocol specifying a format of a data area ;

issuing an alarm when the transport level protocol is identified as User Datagram Protocol and the data segment associated with User Datagram Protocol packet contains two or less bytes of data.



PACKET HEADERS

FIG. 2

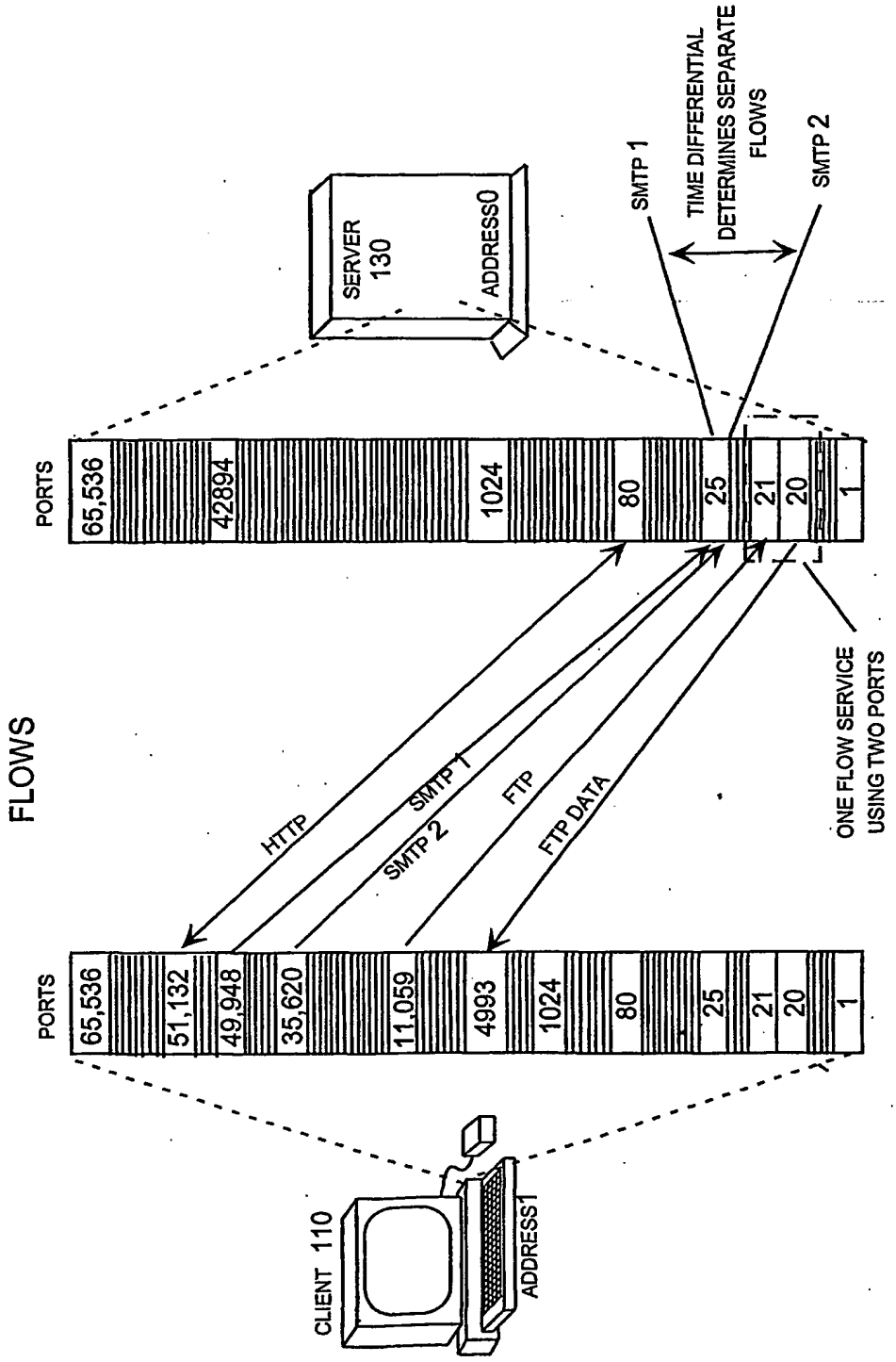

SERVER 130
ADDRESS0
SMTP 1
TIME DIFFERENTIAL DETERMINES SEPARATE FLOWS
SMTP 2
PORTS
65,536
42894
1024
80
25
21
20
1
ONE FLOW SERVICE USING TWO PORTS
FLOWS
HTTP
SMTP 1
SMTP 2
FTP
FTP DATA
PORTS
65,536
51,132
49,948
35,620
11,059
4993
1024
80
25
21
20
1
CLIENT 110
ADDRESS1

FIG. 4

TABLE I

NAME	POTENTIAL INTRUDER	RESPONSE	CI VALUE
POTENTIAL TCP PROBE	TCP PACKETS	RESET PACKETS	NUMBER OF PACKETS
POTENTIAL UDP PROBE	UDP PACKEST	ICMP PORT UNAVAILABLEPCKETS	NUMBER OF ICMP PORT UNAVAILABLE PACKETS
HALF-OPEN ATTACK	HIGH NUMBER AND RATE OF SYNS	SYN-ACKS	5000+501 PER SYN-ACK
TCP STEALTH PORT SCAN	MULTIPLE PACKETS FROM SAME SOURCE PORT TO DIFFERENT DESTINATION PORTS	RESETS	8000+1010 PER PORT OVER 4
UDP STEALTH PORT SCAN	MULTIPLE PACKETS FROM SAME SOURCE PORT TO DIFFERENT DESTINATION PORTS	NOTHING OR ICMP PORT UNAVAILABLE	8000+1010 PER PORT OVER 4

FLOW-BASED CI VALUES

FIG. 6

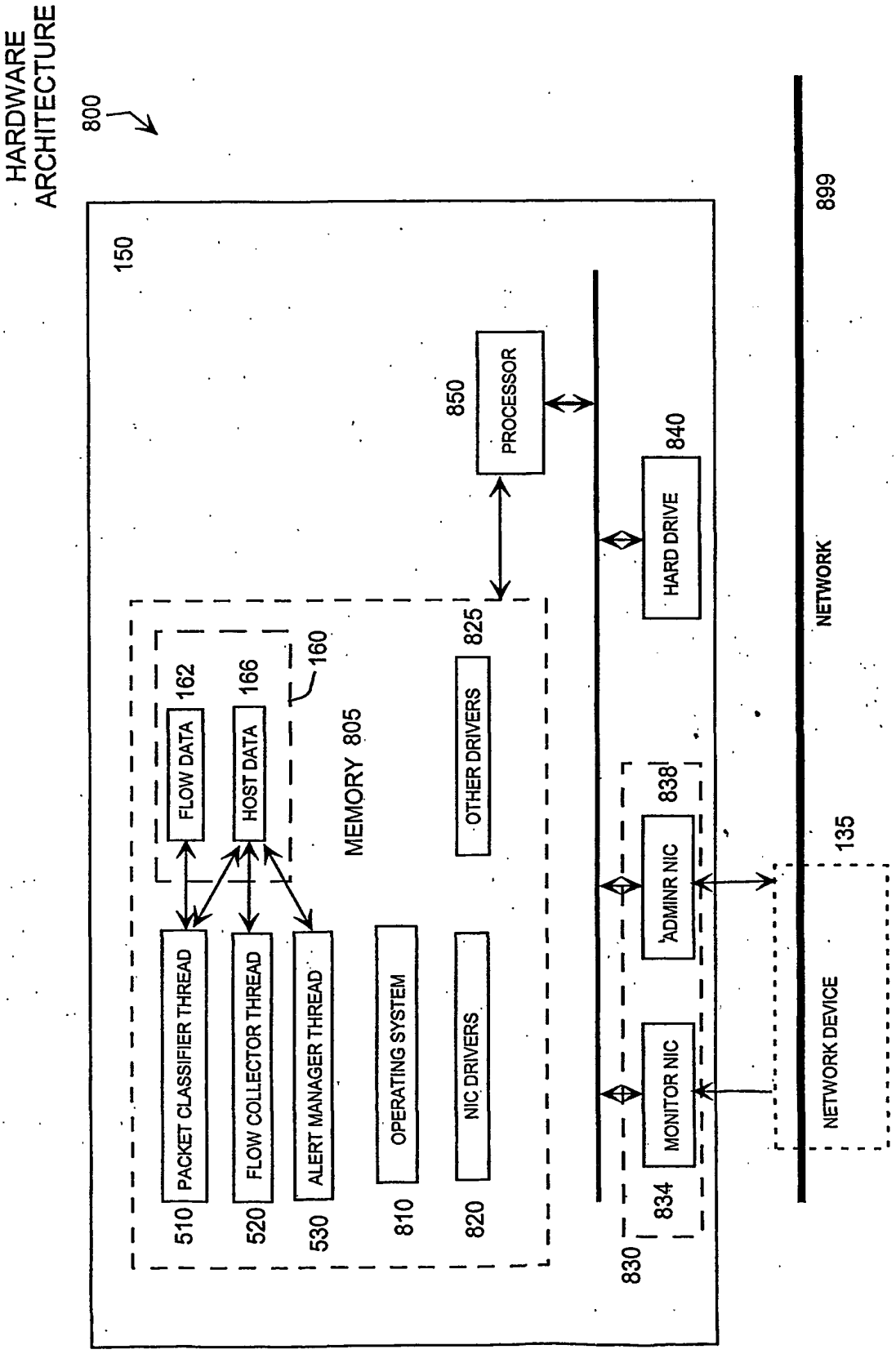

HARDWARE ARCHITECTURE
800
150
899
NETWORK
850
PROCESSOR
840
HARD DRIVE
160
MEMORY 805
162
FLOW DATA
166
HOST DATA
825
OTHER DRIVERS
838
ADMINR NIC
135
NETWORK DEVICE
510
PACKET CLASSIFIER THREAD
520
FLOW COLLECTOR THREAD
530
ALERT MANAGER THREAD
810
OPERATING SYSTEM
820
NIC DRIVERS
830
834
MONITOR NIC

FIG. 8

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 June 2002 (06.06.2002)

PCT

(10) International Publication Number
WO 02/045380 A3

(51) International Patent Classification[7]: H04L 29/06, G06F 1/00

(21) International Application Number: PCT/US01/45275

(22) International Filing Date:
30 November 2001 (30.11.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/250,261 30 November 2000 (30.11.2000) US
60/265,194 31 January 2001 (31.01.2001) US

(71) Applicant *(for all designated States except US)*: LANCOPE, INC. [US/US]; 1070 Greenway, Atlanta, GA 30305 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: COPELAND, John, A. III [US/US]; 1070 Greenway, Atlanta, GA 30305 (US).

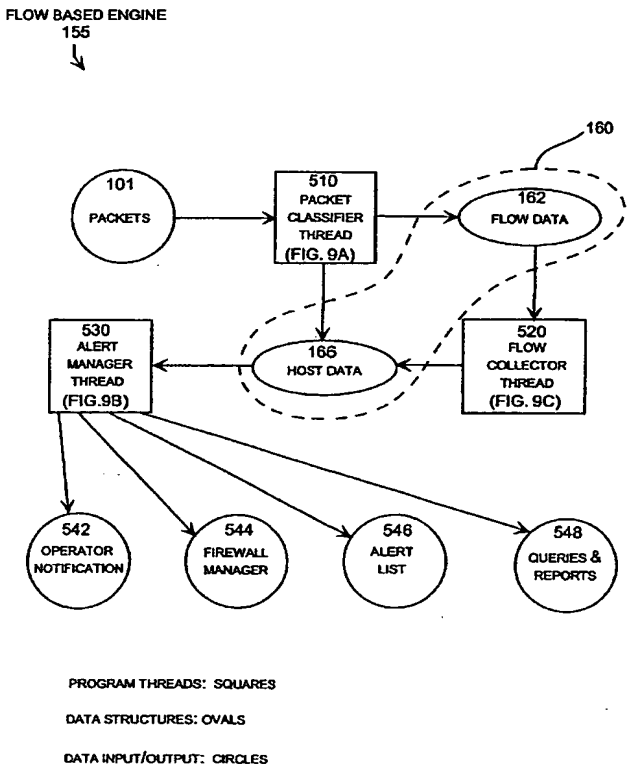
(74) Agent: HARRIS, John, R.; Morris, Manning & Martin, LLP, 1600 Atlanta Financial Center, 3343 Peachtree Road, N.E., Atlanta, GA 3032601944 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: FLOW-BASED DETECTION OF NETWORK INTRUSIONS



(57) Abstract: A flow-based intrusion detection system for detecting intrusions in computer communication networks. Data packets representing communications between hosts in a computer-to-computer communication network are processed and assigned to various client/server flows. Statistics are collected for each flow. Then, the flow statistics are analyzed to determine if the flow appears to be legitimate traffic or possible suspicious activity. A concern index value is assigned to each flow that appears suspicious. By assigning a value to each flow that appears suspicious and adding that value to the total concern index of the responsible host, it is possible to identify hosts that are engaged in intrusion activity. When the concern index value of a host exceeds a preset alarm value, an alert is issued and appropriate action can be taken.

WO 02/045380 A3

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/45275

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	WO 00 34847 A (DIEP THANH A ;VISA INT SERVICE ASS (US)) 15 June 2000 (2000-06-15) abstract; figures 3-9 --- -/--	1-11 12

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 June 2002	10/07/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Köppl, M

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

ormation on patent family members

International Application No
PCT/US 01/45275

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0034847 A	15-06-2000	US 6370648 B1	09-04-2002
		AU 2046400 A	26-06-2000
		EP 1137976 A2	04-10-2001
		WO 0034847 A1	15-06-2000
US 6321338 B1	20-11-2001	NONE	
WO 0131420 A	03-05-2001	AU 2903901 A	08-05-2001
		WO 0131420 A2	03-05-2001